LV24-33 v6™

User manual

All MikroElektronika´s development systems represent irreplaceable tools for programming and developing microcontroller-based devices. Carefully chosen components and the use of machines of the last generation for mounting and testing thereof are the best guarantee of high reliability of our devices. Due to simple design, a large number of add-on modules and ready to use examples, all our users, regardless of their experience, have the possibility to develop their project in a fast and efficient way.

Development System

MikroElektronika

SOFTWARE AND HARDWARE SOLUTIONS FOR EMBEDDED WORLD ...making it simple

TO OUR VALUED CUSTOMERS

I want to express my thanks to you for being interested in our products and for having confidence in mikroElektronika.
The primary aim of our company is to design and produce high quality electronic products and to constantly improve the performance thereof in order to better suit your needs.

Nebojsa Matic
General Manager

The Microchip name and logo, the Microchip logo, Accuron, dsPIC, KeeLoq, microID, MPLAB, PIC, PICmicro, PICSTART, PRO MATE, PowerSmart, rfPIC and SmartShunt are registered trademarks of Microchip Technology Incorporated in the U.S.A and other countries.

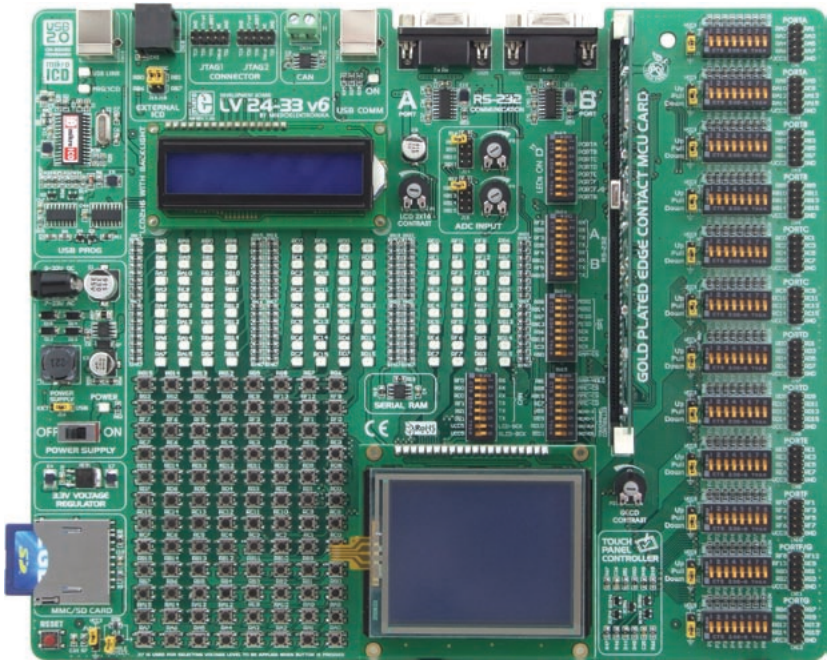
TABLE OF CONTENTS

Introduction to LV24-33 v6 Development System 4
Key Features 5
1.0. Connecting the System to a PC 6
2.0. Supported Microcontrollers 7
3.0. On-board LvPICFlash Programmer 9
4.0. MikroICD (In-Circuit Debugger) 10
5.0. ICD Connector 11
6.0. JTAG Connector 12
7.0. Power Supply 13
8.0. 3.3V Voltage Regulator 14
9.0. USB Communication Connector 14
10.0. RS-232 Communication Module 15
11.0. CAN Communication Module 16
12.0. A/D Converter Test Inputs 17
13.0. MMC/SD Connector 18
14.0. RAM Module 19
15.0. LEDs 20
16.0. Push Buttons 21
17.0. 2x16 LCD 22
18.0. 128x64 Graphic LCD 23
19.0. Touch Panel 24
20.0. I/O Ports 25

page

Introduction to LV24-33 v6 Development System

The *LV24-33 v6*™ development system provides a development environment for experimenting with dsPIC® microcontrollers from Microchip®. The system includes an on-board programmer providing an interface between the microcontroller and a PC. You are simply expected to write a program in one of the dsPIC compilers, generate a .hex file and program your microcontroller using the on-board *LvPICFlash*™ programmer. Numerous modules, such as 128x64 graphic LCD, alphanumeric 2x16 LCD, serial RAM etc., are provided on the board and allow you to easily simulate the operation of the target device.



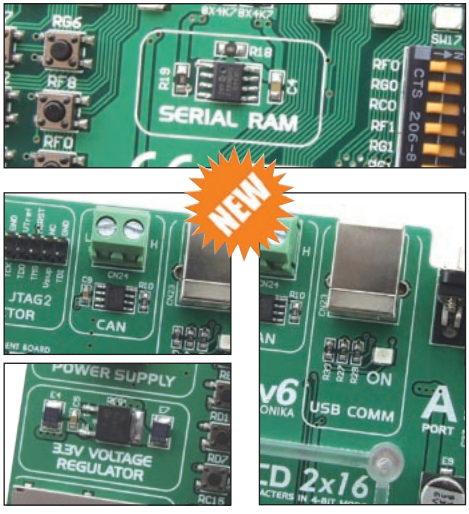
LV24-33 DEVELOPMENT BOARD
Full-featured development system for dsPIC microcontroller based devices

USB 2.0 IN-CIRCUIT PROGRAMMER
USB 2.0 on-board programmer

TOUCHSCREEN
A possibility of connecting a graphic display with touch panel increases the functionality of the development system

MMC/SD MEMORY CARD
A possibility of reading MMC/SD memory cards

GLCD128x64
Graphic LCD with backlight



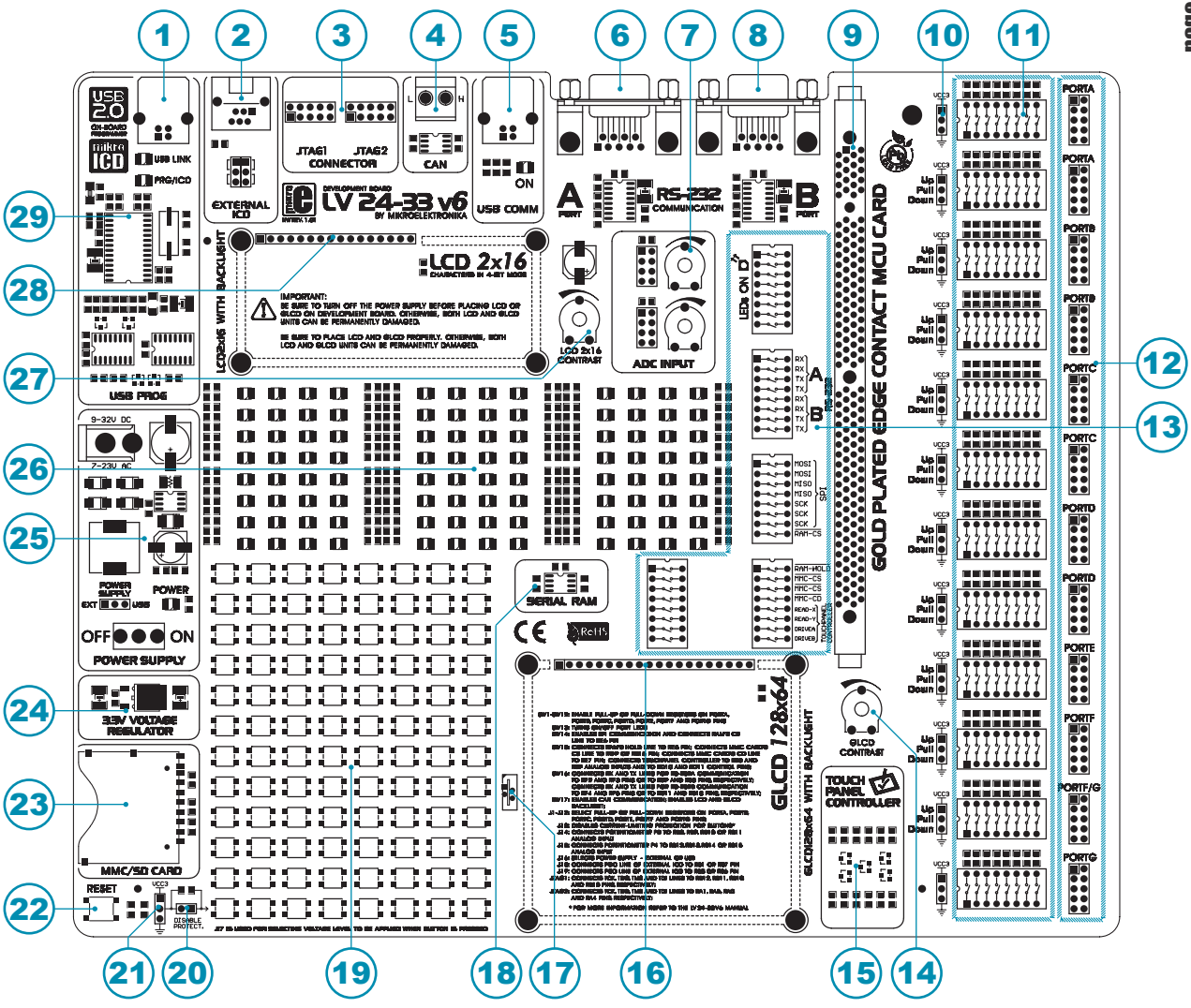
The *LvPICFLASH*™ program for programming provides a complete list of all supported microcontrollers. The latest version of this program with updated list of supported microcontrollers can be downloaded from our website at **www.mikroe.com**

Package contains:

Development system: LV24-33 v6
CD: product CD with relevant software
Cables: USB cable
Documentation: Manuals for LV24-33 v6 and mikroICD, quick guide for installing USB drivers, electrical schematic of the system

System specification:

Power Supply: over an AC/DC connector (7-23V AC or 9-32V DC); or over a USB cable for programming (5V DC)
Power consumption: ~40mA in idle state when all on-board modules are off
Dimension: 26,5 x 22cm (10,4 x 8,6inch)
Weight: ~420g (0.92lbs)



Key Features

1. On-board programmer's USB connector
2. Connector for external programmer ICD2 or ICD3
3. JTAG connectors
4. CAN communication module
5. USB communication connector
6. Serial communication connector RS-232A
7. A/D converter test inputs
8. Serial communication connector RS-232B
9. DIMM 168P socket for MCU cards
10. Jumper for pull-up/pull-down resistor selection
11. DIP switch enables pull-up/pull-down resistors
12. I/O port connectors
13. DIP switch turns on/off on-board modules
14. Potentiometer for adjusting contrast of graphic display
15. Touch panel controller
16. Graphic LCD connector
17. Touch panel connector
18. Serial RAM module
19. Push buttons simulate microcontroller digital inputs
20. Jumper for protective resistor shortening
21. Jumper for selecting push buttons' logic state
22. Reset button
23. MMC/SD card connector
24. 3.3V voltage regulator
25. Power supply voltage regulator
26. 96 LEDs indicate pins' logic state
27. Potentiometer for adjusting contrast of LCD
28. Alphanumeric LCD display connector
29. On-board programmer with mikroICD support

page

1.0. Connecting the System to a PC

Step 1:

Follow the instructions provided in the relevant manuals and install the *LvPICFLASH* program and USB drivers from the product CD. USB drivers are essential for the proper operation of the on-board programmer.
In case you already have one of the Mikroelektronika's dsPIC compilers installed on your PC, there is no need to reinstall USB drivers as they are already installed along with the compiler.

Step 2:

Use the USB cable to connect the USB development system to a PC. One end of the USB cable, with a USB connector of B type, should be connected to the development system, as shown in Figure 1-2, whereas the other end of the cable with a USB connector of A type should be connected to a PC. When establishing a connection, make sure that jumper J16 is placed in the USB position as shown in Figure 1-1.

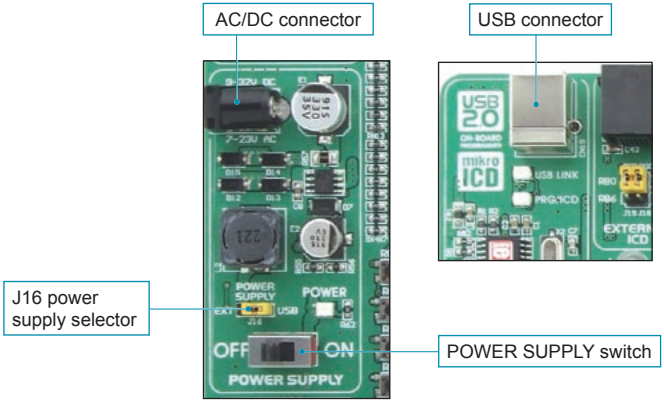


Figure 1-1: Power supply

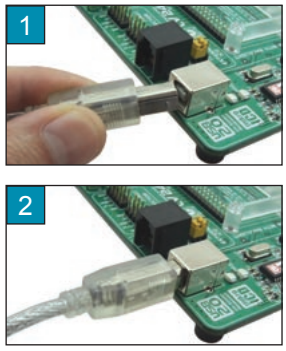


Figure 1-2: Connecting USB cable

Step 3:

Turn on your development system by setting the POWER SUPPLY switch to the ON position. Two LEDs marked as POWER and USB LINK will be automatically turned on indicating that your development system is ready to use. Use the on-board programmer and the *LvPICFLASH* program to dump a code into the microcontroller and employ the system to test and develop your projects.

NOTE: If some additional modules are used, such as LCD, GLCD etc, it is necessary to place them properly on the development system while it is turned off. Otherwise, either can be permanently damaged. Refer to figure below for the proper placing of the additional modules.

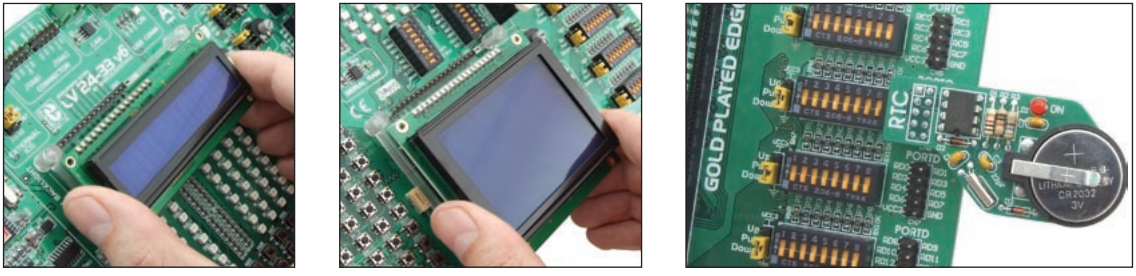


Figure 1-3: Placing additional modules on the board

2.0. Supported Microcontrollers

The *LV24-33 v6* development system provides a DIMM-168P connector to place an MCU card into. This development system comes with an MCU card with the PIC24FJ96GA010 microcontroller in 100-pin TQFP package soldered on it, Figure 2-3. Besides, the MCU card alone provides an oscillator as well as 102 soldering pads connected to microcontroller pins. Each pad is marked the same as the pin it is connected to. Soldering pads also make connection between the MCU card and target device's modules easy.

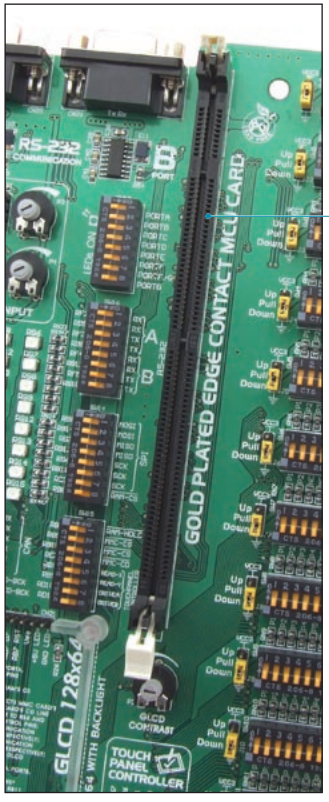


Figure 2-1: DIMM-168P connector

DIMM-168P connector for placing the MCU card with microcontroller in TQFP package

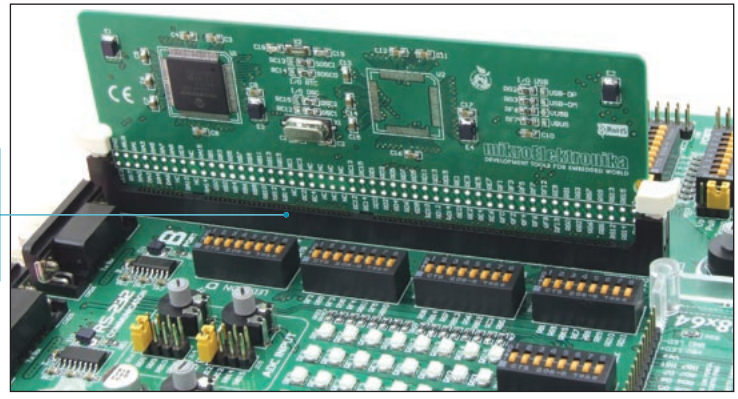


Figure 2-2: DIMM-168P connector with the MCU card inserted

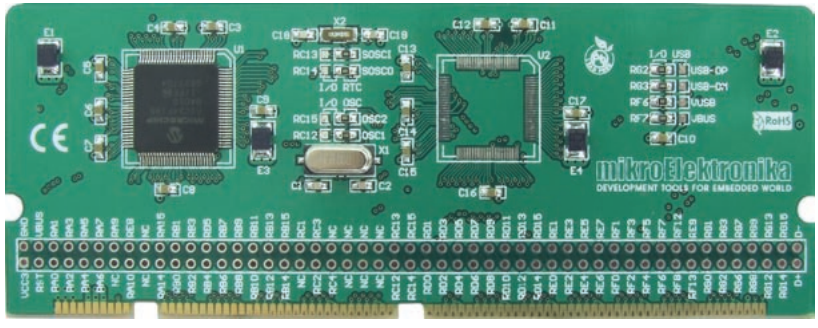


Figure 2-3: MCU card with a 100-pin microcontroller in TQFP package soldered on it

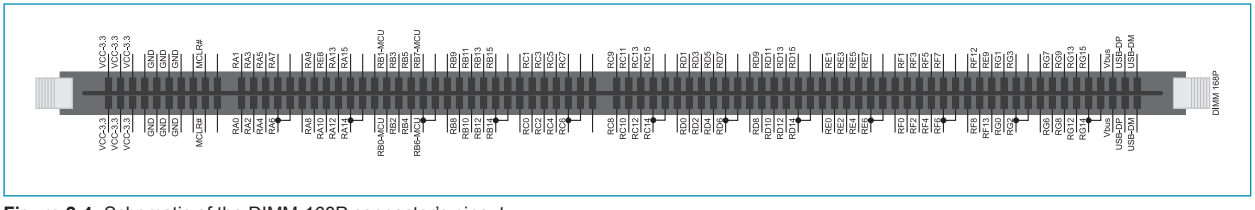
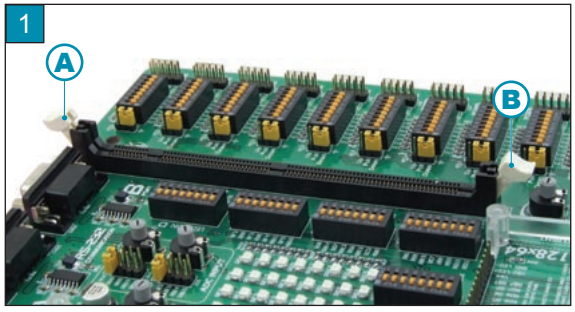


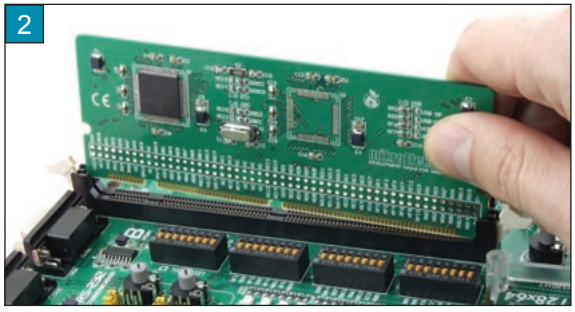
Figure 2-4: Schematic of the DIMM-168P connector's pinout

page

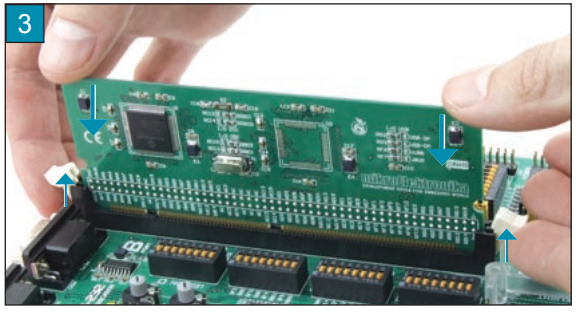
Placing MCU card into the DIMM-168P connector is performed as follows:



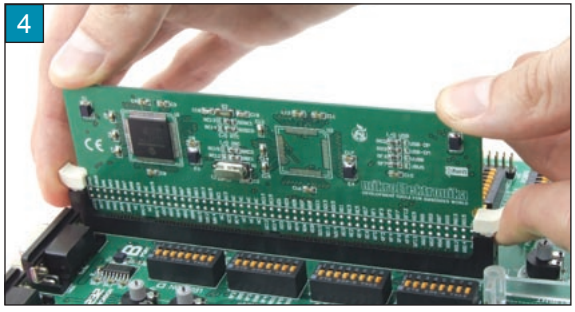
Open extraction levers A and B



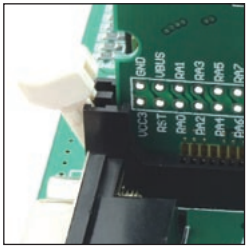
Place the MCU card into the DIMM-168P connector

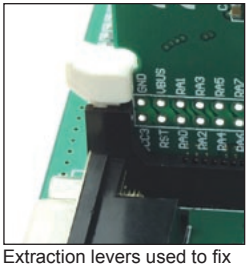


Push the MCU card down gently into the DIMM-168P connector and lift extraction levers slowly at the same time



Close the extraction levers when the MCU card is properly placed into the connector


Extraction levers used to fix the MCU card in ‘open’ position


Extraction levers used to fix the MCU card in ‘closed’ position

In addition to the MCU card with microcontrollers in 100-pin TQFP package, there are also MCU cards with microcontrollers in 44, 64 and 80-pin TQFP package which can be ordered separately. They are placed into the appropriate connector in the same manner as described above.

3.0. On-board LvPICFlash Programmer

A programmer is a necessary tool when working with microcontrollers. It is used to load a hex code into the microcontroller and provides an interface between the microcontroller and the PC. The *LV24-33 v6* has an on-board *LvPICFlash* programmer. The *LvPICFLASH* program is used for loading a .hex file into the microcontroller. Figure 3-3 shows connection between the compiler, *LvPICFLASH* program and the microcontroller.

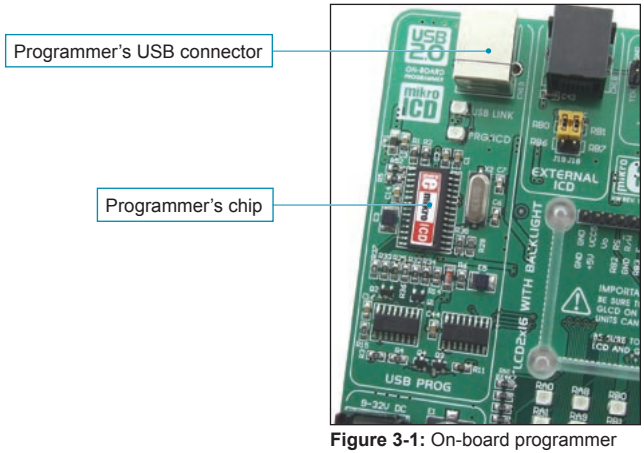


Figure 3-1: On-board programmer

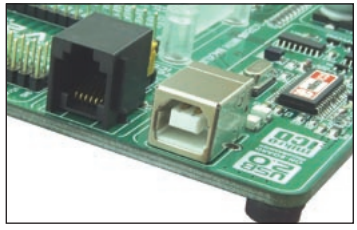


Figure 3-2: USB connector's front side

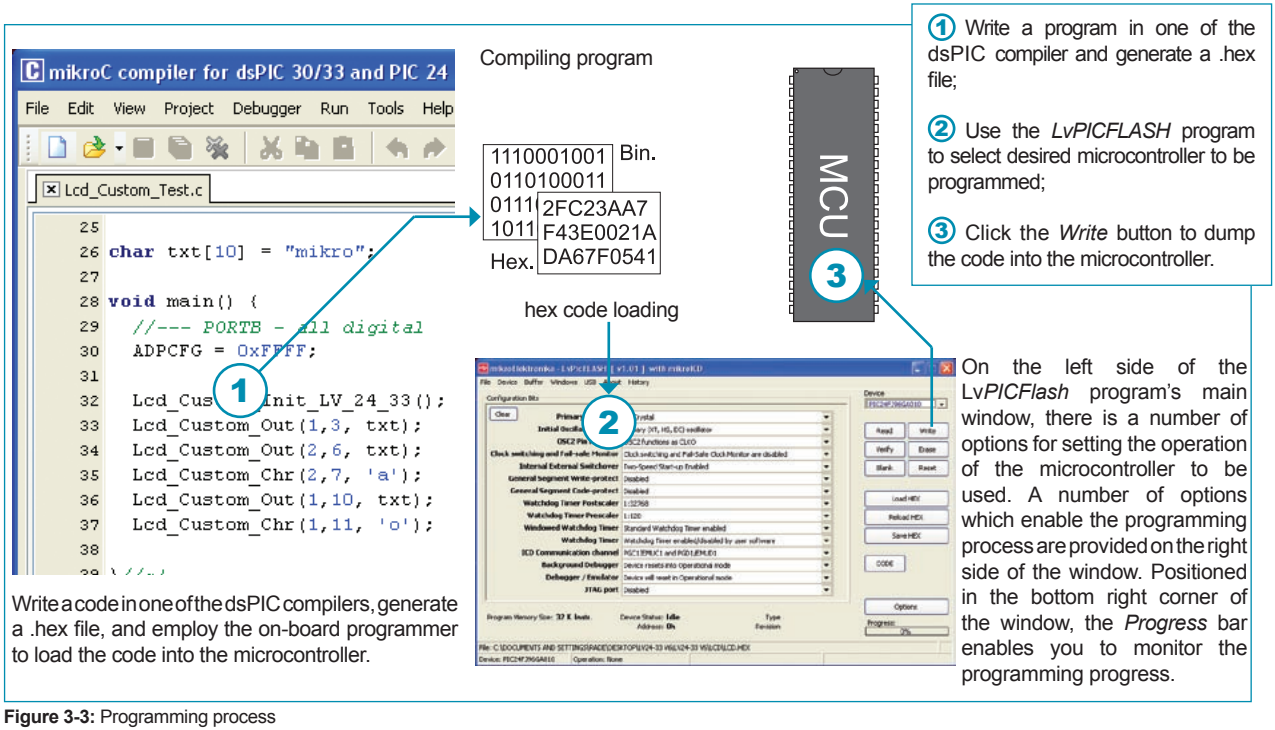

mikroC compiler for dsPIC 30/33 and PIC 24
File Edit View Project Debugger Run Tools Help
Lcd_Custom_Test.c
char txt[10] = "mikro";
void main() {
//--- PORTB - all digital
ADPCFG = 0xFFFF;
Lcd_Custom_Init_LV_24_33();
Lcd_Custom_Out(1,3, txt);
Lcd_Custom_Out(2,6, txt);
Lcd_Custom_Chr(2,7, 'a');
Lcd_Custom_Out(1,10, txt);
Lcd_Custom_Chr(1,11, 'o');
Compiling program
1110001001 0110100011 0111 1011
Bin.
2FC23AA7 F43E0021A DA67F0541
Hex.
hex code loading
MCU
1
2
3
① Write a program in one of the dsPIC compiler and generate a .hex file;
② Use the LvPICFLASH program to select desired microcontroller to be programmed;
③ Click the Write button to dump the code into the microcontroller.
Write a code in one of the dsPIC compilers, generate a .hex file, and employ the on-board programmer to load the code into the microcontroller.
On the left side of the LvPICFlash program's main window, there is a number of options for setting the operation of the microcontroller to be used. A number of options which enable the programming process are provided on the right side of the window. Positioned in the bottom right corner of the window, the Progress bar enables you to monitor the programming progress.

Figure 3-3: Programming process

page

4.0. mikroICD (In-Circuit Debugger)

The mikroICD (In-Circuit Debugger) is an integral part of the on-board programmer. It is used for the purpose of testing and debugging programs in real time. The process of testing and debugging is performed by monitoring the state of all registers within the microcontroller while operating in real environment. The mikroICD software is integrated in all dsPIC compilers designed by Mikroelektronika (mikroBASIC PRO, mikroC PRO, mikroPASCAL PRO etc). As soon as the mikroICD debugger starts up, a window called *Watch Values*, appears on the screen, Figure 4-1. The *mikroICD* debugger communicates to the microcontroller through the microcontroller's pins used for programming.

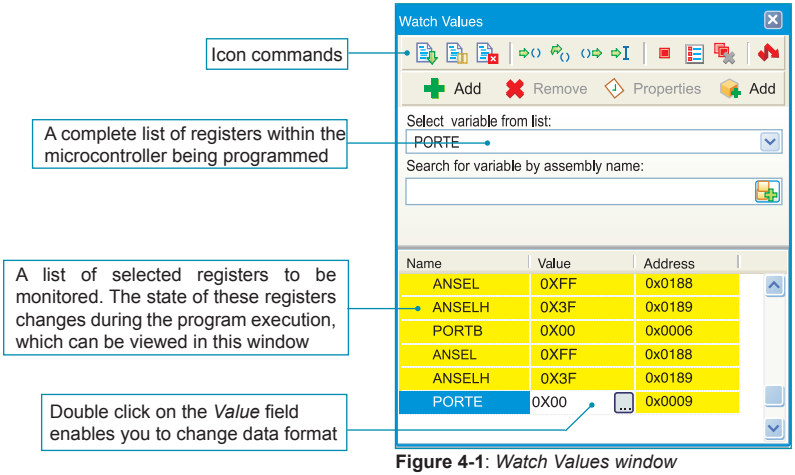

Icon commands
A complete list of registers within the microcontroller being programmed
A list of selected registers to be monitored. The state of these registers changes during the program execution, which can be viewed in this window
Double click on the Value field enables you to change data format
Watch Values
Add
Remove
Properties
Add
Select variable from list:
PORTE
Search for variable by assembly name:
Name
Value
Address
ANSEL 0XFF 0x0188
ANSELH 0X3F 0x0189
PORTB 0X00 0x0006
ANSEL 0XFF 0x0188
ANSELH 0X3F 0x0189
PORTE 0X00 0x0009

Figure 4-1: *Watch Values window*

mikroICD debugger options:

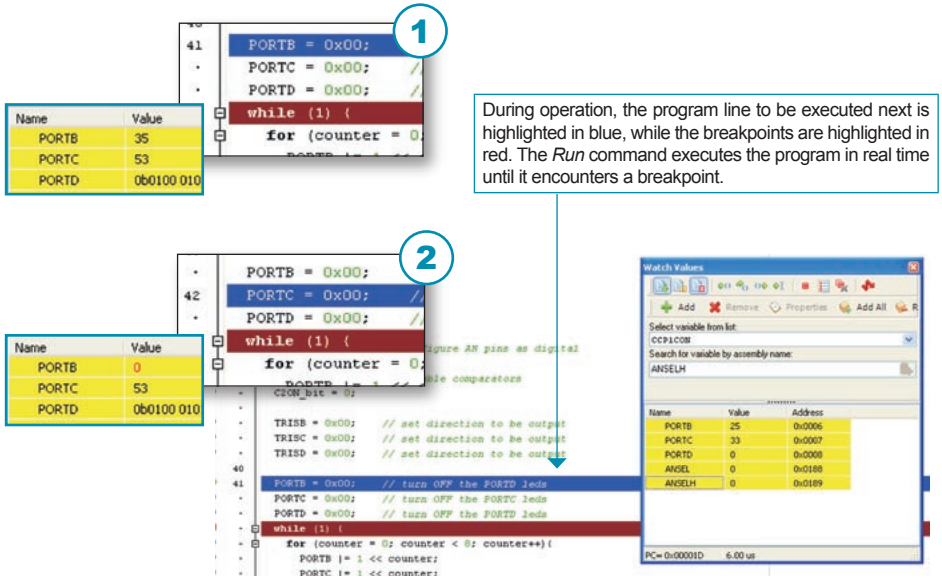
Start Debugger	[F9]
Run/Pause Debugger	[F6]
Stop Debugger	[Ctrl+F2]
Step Into	[F7]
Step Over	[F8]
Step Out	[Ctrl+F8]
Toggle Breakpoint	[F5]
Show/Hide Breakpoints	[Shift+F4]
Clear Breakpoints	[Ctrl+Shift+F5]

Each of these commands is activated via keyboard shortcuts or by clicking appropriate icon within the *Watch Values* window.

The mikroICD debugger also offers functions such as running a program step by step (single stepping), pausing the program execution to examine the state of currently active registers using breakpoints, tracking the values of some variables etc. The following example illustrates a step-by-step program execution using the *Step Over* command.

Step 1:
In this example the 41st program line is highlighted in blue, which means that it will be executed next. The current state of all registers within the microcontroller can be viewed in the mikroICD *Watch Values* window.

Step 2:
After the *Step Over* command is executed, the microcontroller will execute the 41st program line. The next line to be executed is highlighted in blue. The state of registers being changed by executing this instruction may be viewed in the *Watch Values* window.


1
2
PORTB = 0x00;
PORTC = 0x00;
PORTD = 0x00;
while (1) {
for (counter = 0;
Name Value
PORTB 35
PORTC 53
PORTD 0b0100 010
Name Value
PORTB 0
PORTC 53
PORTD 0b0100 010
During operation, the program line to be executed next is highlighted in blue, while the breakpoints are highlighted in red. The Run command executes the program in real time until it encounters a breakpoint.
Watch Values

NOTE: For more information on the mikroICD debugger refer to the *mikroICD Debugger* manual.

5.0. ICD Connector

The ICD connector enables communication between the microcontroller and an external ICD debugger/programmer from Microchip (ICD2® or ICD3®). Jumpers J18 and J19 are used for selecting the pin to be fed with programming signal. For the programming signal to be sent to the microcontroller, it is necessary to place both jumpers in the same position, as shown in Figure 5-3. Either RB0 and RB1 or RB6 and RB7 microcontroller pins can be fed with this signal. The position of jumpers J18 and J19 depends on the microcontroller provided on the MCU card. When the PIC24FJ96GA010 microcontroller is programmed by means of the external ICD programmer, jumpers J18 and J19 can be placed in any position. The main reason for it is the capability of the microcontroller to be programmed over one of two available pairs of programming pins (RB0 and RB1 or RB6 and RB7).

The ICD3 programmer cannot be used for programming the microcontroller until the power supply is off. In this case it is necessay to enable the development system to be powered over the ICD3 programmer. The power supply should not be suspended when using the ICD2 programmer. Never use the ICD2 programmer for powering the system as it may cause the microcontroller to be permanently damaged. The reason for it is the 5V power supply voltage provided by the ICD2 programmer, which is destructive for the microcontroller which employs the 3.3V power supply voltage.

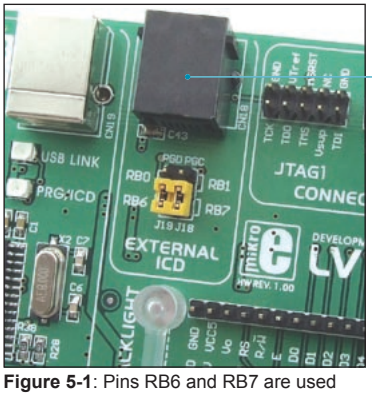
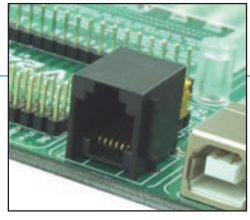


Figure 5-1: Pins RB6 and RB7 are used for programming



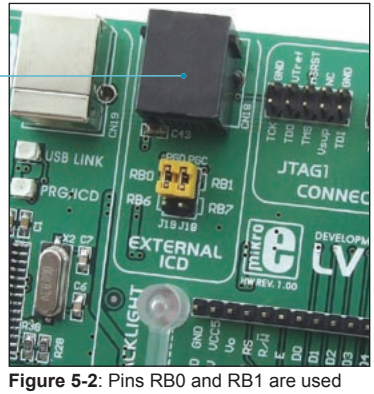


Figure 5-2: Pins RB0 and RB1 are used for programming

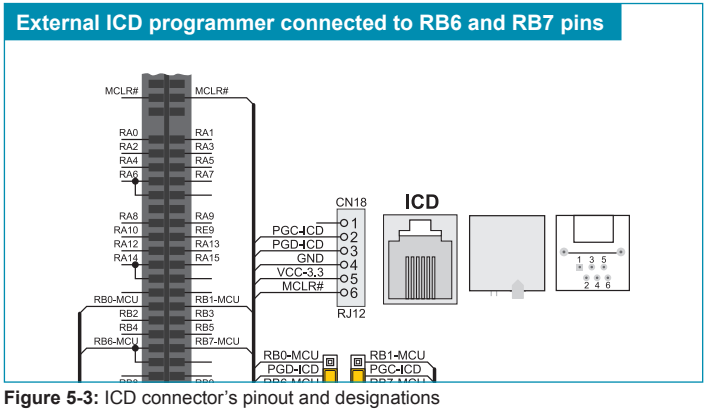


Figure 5-3: ICD connector's pinout and designations

page

6.0. JTAG Connector

JTAG is a programmer/debugger used for programming/debugging microcontrollers provided with a built-in JTAG interface. This JTAG interface is a modified version of the original JTAG interface and enables the contents of internal EEPROM and FLASH memory to be altered (programming microcontrollers). JTAG connectors (JTAG1 and JTAG2) are directly connected to microcontroller pins. The JTAG1 connector is linked to RB10, RB11, RB12, RB13 and MCLR# pins. The JTAG2 connector is linked to RA0, RA1, RA4, RA5 and MCLR# pins.

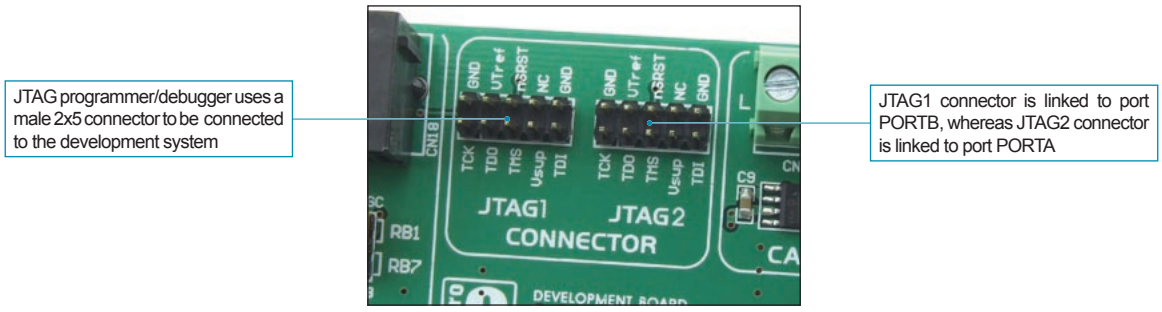


Figure 6-1: JTAG connector

JTAG connectors are directly connected to microcontroller pins

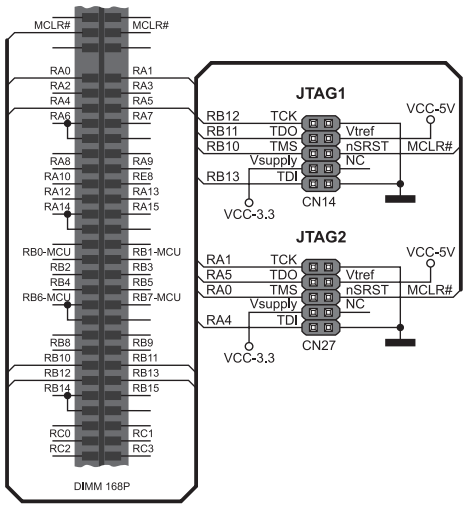


Figure 6-2: JTAG connector connection schematic

7.0. Power supply

The *LV24-33 v6* development system may use one of two power supply sources:

1. +5V PC power supply through the USB programming cable; and
2. External power supply source connected to an AC/DC connector provided on the development board.

The MC34063A voltage regulator and Gretz rectifier are used to enable external power supply voltage to be either AC (in the range of 7V to 23V) or DC (in the range of 9V to 32V). Jumper J16 is used as a selector for a power supply source. To make advantage of the USB power supply, jumper J16 should be placed in the USB position. When using the external power supply, jumper J16 should be placed in the EXT position. The development system is turned on/off by switching the position of the POWER SUPPLY switch.

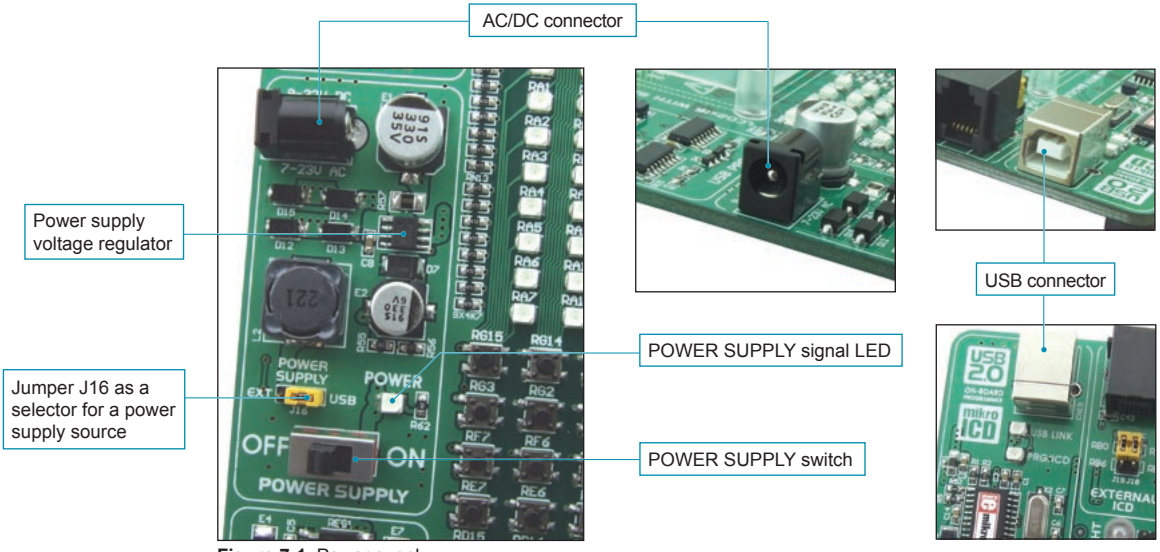


Figure 7-1: Power supply

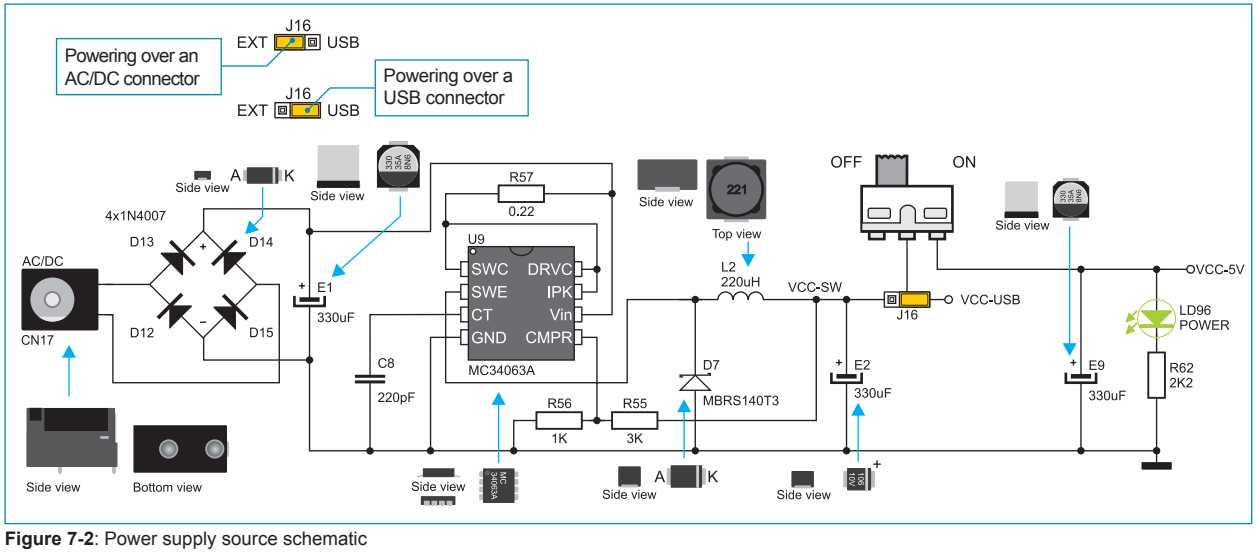


Figure 7-2: Power supply source schematic

page

8.0. 3.3V Voltage Regulator

The *LV24-33 v6* development system is provided with a voltage regulator whose purpose of which is to reduce the power supply voltage of the development system from 5V to 3.3V. The 3.3V power supply voltage is then used for powering the microcontroller and other on-board modules. The MC33269DT-3.3 circuit provided on the development system is used as a voltage regulator capable of providing 800mA current.

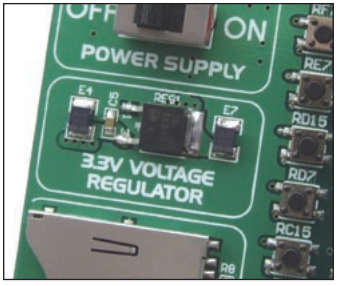


Figure 8-1: Voltage regulator

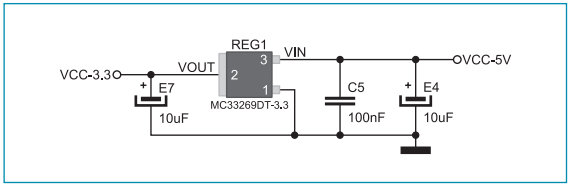


Figure 8-2: Voltage regulator connection schematic

9.0. USB Communication Module

The CN23 USB connector enables a dsPIC with a built-in USB interface to be connected to peripheral devices. The microcontroller is connected to the CN23 connector via USB-DM and USB-DP lines. The function of the LED marked as ON is to indicate connection between USB devices and development system, hence the microcontroller.

Figure 9-1: USB connector

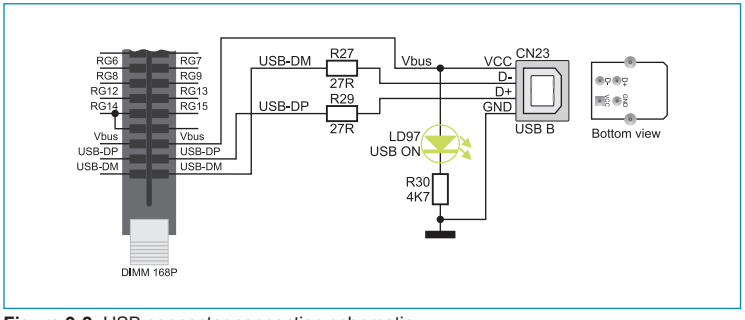


Figure 9-2: USB connector connection schematic

10.0. RS-232 Communication Module

USART (*Universal Synchronous/Asynchronous Receiver/Transmitter*) is one of the most common ways of exchanging data between the PC and peripheral units. RS-232 serial communication is performed through a 9-pin SUB-D connector and the microcontroller USART module. The *LV24-33 v6* provides two RS-232 ports, RS-232A and RS-232B. Use switches marked as RX232-A (pin RF2 or RB9) and TX232-A (pin RF3 or RB8) on the DIP switch SW16 to enable port RS-232A. Likewise, use switches marked as RX232-B (pin RF4 or RB11) and TX232-B (pin RF5 or RB10) on the DIP switch SW16 to enable port RS-232B. The microcontroller pins used in such communication are marked as follows: RX - *receive data line* and TX - *transmit data line*. Data rate goes up to 115 kbps.

In order to enable the USART module of the microcontroller to receive input signals which meet the RS-232 standard, it is necessary to adjust voltage levels using an IC circuit such as MAX3232CDR.

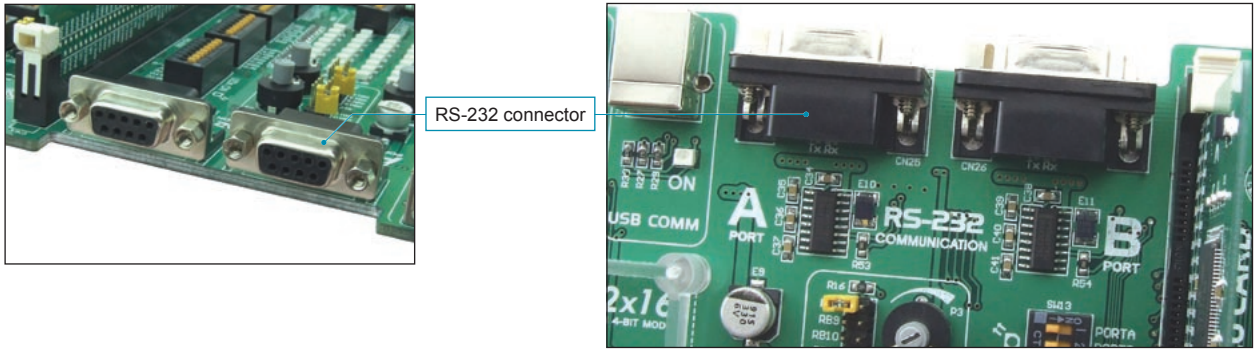


Figure 10-1: RS-232 module

The function of the switches 1-8 on the DIP switch SW16 is to determine which of the microcontroller pins are to be used as RX and TX lines, Figure 10-2.

Port RS-232A is connected to the microcontroller via RF2 and RF3 pins

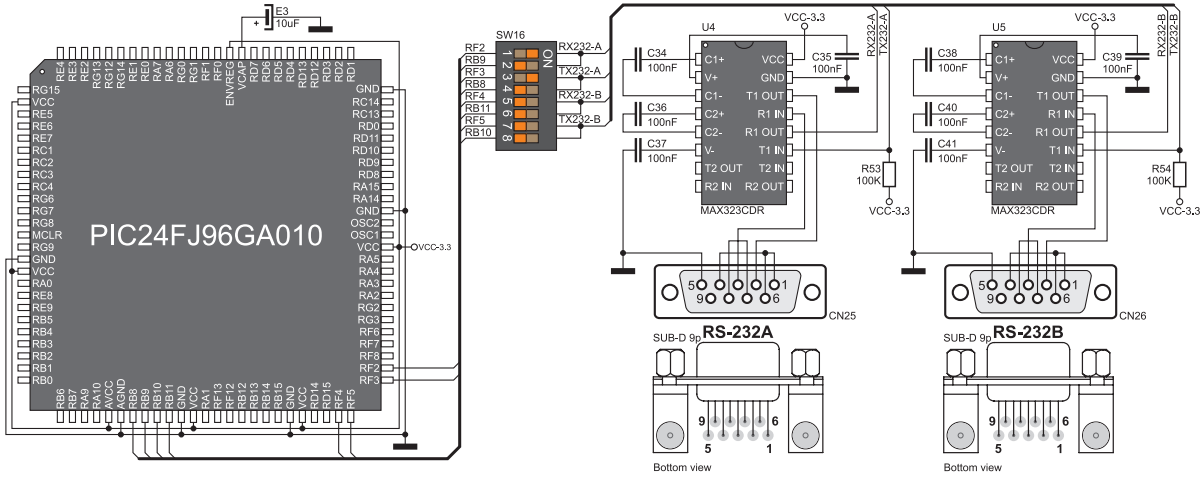


Figure 10-2: RS-232 module connection schematic

page

11.0. CAN Communication Module

CAN (Controller Area Network) is a communication standard primarily intended for use in automotive industry. It enables the microcontroller to communicate to a car device without using a host PC. In addition, such communication is widely used in industrial automation. The *LV24-33 v6* uses the MCP2551 circuit for CAN communication. This circuit provides an interface between the microcontroller and some peripheral device. To enable connection between the microcontroller and MCP2551, it is necessary to set switches 1 and 4 on the DIP switch SW17 to ON position. Other switches (1-6) on the DIP switch SW17 may also be used for this purpose. Which of these switches is to be used depends solely on the arrangement of the microcontroller pins connected to CAN communication lines. RX lines are connected to the microcontroller via RF0, RG0 and RC0 microcontroller pins. TX line is connected to the microcontroller via RF1, RG1 and RC1 microcontroller pins.

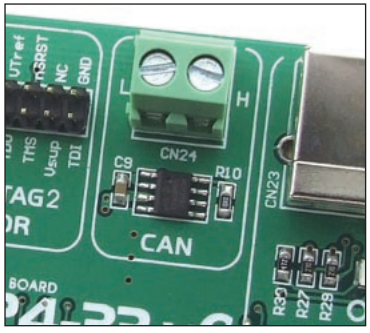


Figure 11-1: CAN module

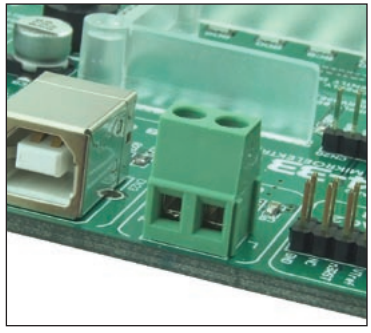


Figure 11-2: CAN module connector

CAN communication is enabled via DIP switch SW17

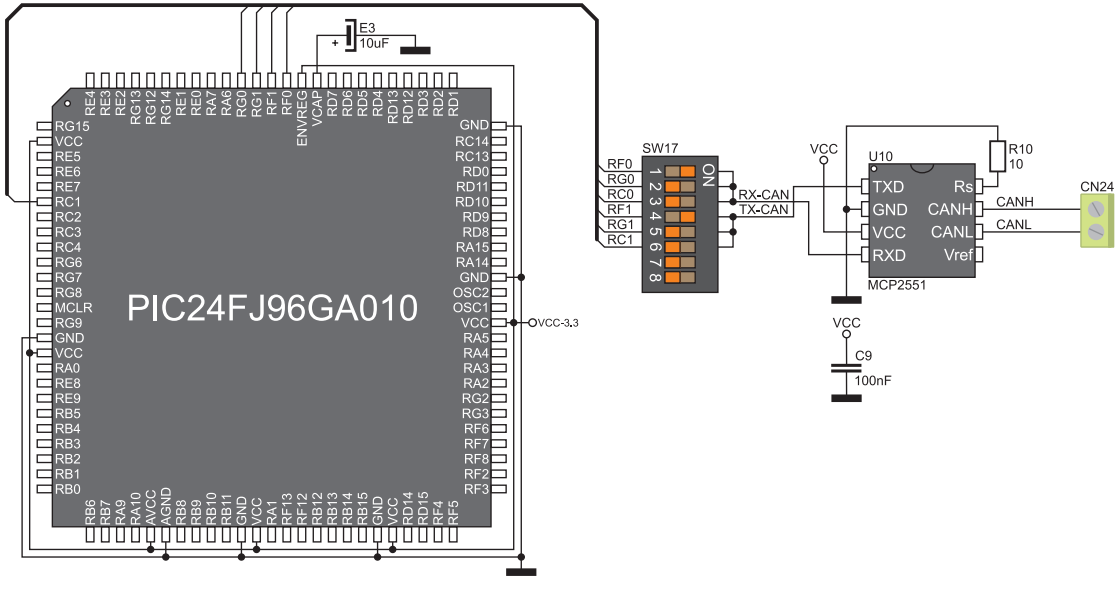


Figure 11-3: Microcontroller and MCP2551 connection schematic

12.0. A/D Converter Test Inputs

An A/D converter is used for converting an analog voltage into the appropriate digital value. The A/D converter is linear, which means that converted number is linearly dependent on the input voltage value. The A/D converter built into the microcontroller converts an analog voltage value into a 10-bit number. Potentiometers P3 and P4 enables voltage to vary between 0 and 3.3V. The microcontroller with a built-in A/D converter is supplied with this voltage via test inputs. Jumpers J14 and J15 are used for selecting one of the following pins (RB8 - RB15) to be supplied with A/D conversion voltage. Resistors R16 and R17 have a protective function and are used to limit current flow through the potentiometer or the microcontroller pin.

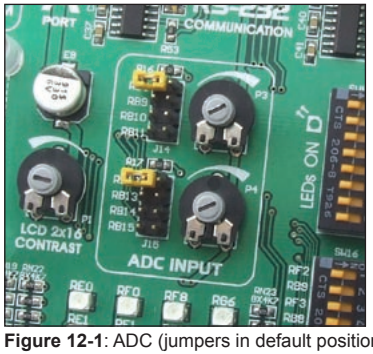


Figure 12-1: ADC (jumpers in default position)

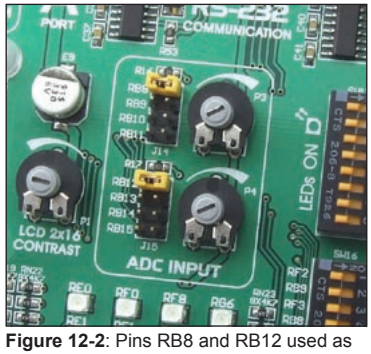


Figure 12-2: Pins RB8 and RB12 used as input pins for A/D conversion

A/D conversion is performed via RB8 and RB12 microcontroller pins

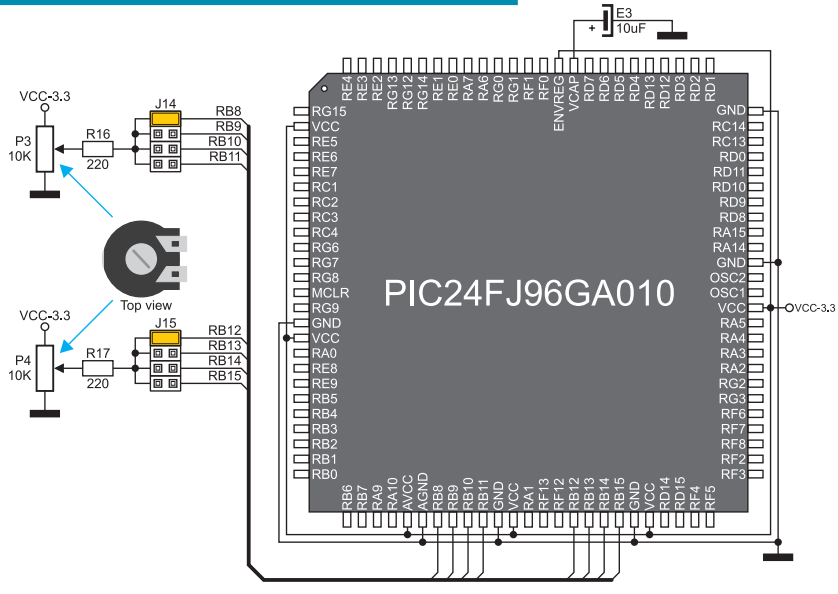


Figure 12-3: Microcontroller and A/D converter test inputs connection schematic

NOTE: In order to enable the microcontroller to accurately perform A/D conversion, it is necessary to turn off LEDs and pull-up or pull-down resistors on the port pins used by the A/D converter.

page

13.0. MMC/SD Connector

The MMC/SD connector enables the memory cards to be interfaced to the microcontroller in order to expand microcontroller memory. To enable serial communication between the microcontroller and a memory card it is necessary to select microcontroller pins to be fed with signal sent from the memory card. DIP switches SW14 and SW15 are used as pin selectors. Figure 13-3 illustrates the connection between the MMC/SD connector and RG9, RE7, RG8, RG7 and RG6 microcontroller pins. Alternatively, pins RB15, RB14, RB13, RB12 and RC3 can be used here as well.

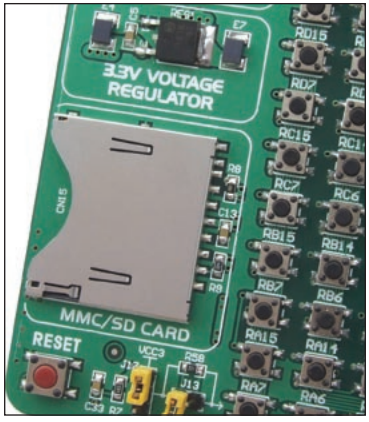


Figure 13-1: MMC/SD connector

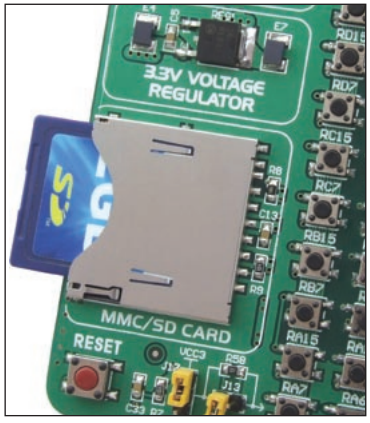


Figure 13-2: MMC/SD memory card

MMC/SD card communicates to the microcontroller via MISO, MOSI, SCK, MMC-CS and MMC-CD lines

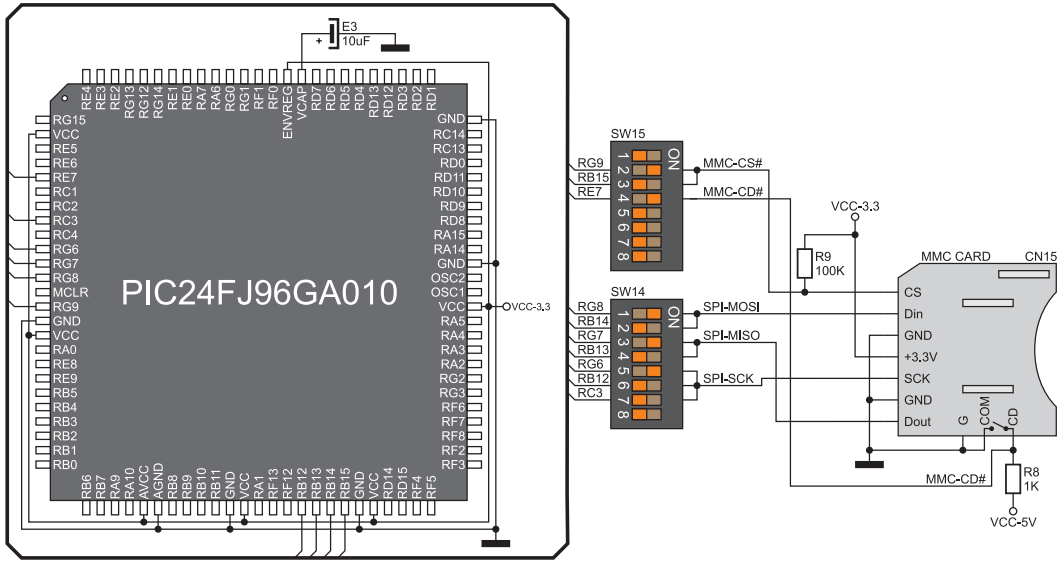


Figure 13-3: MMC/SD connector and microcontroller connection schematic

14.0. RAM Module

Owing to the 23K640 circuit, the development system is provided with 64Kbit RAM modul that can be connected to the microcontroller via SPI interface. The function of the serial RAM module is to expand available RAM of the microcontroller. To establish connection between the microcontroller and RAM module, it is necessary to enable serial communication using DIP switch SW14. In the example in figure 14-2, RAM module is connected to the microcontroller pins RB14, RB13, RB12 and RB0 using DIP switch SW14. Alternatively, RG8, RG7, RG6 and RC3 microcontroller pins may also be used for this purpose. The function of the RB1 (RAM-HOLD#) switch on the DIP switch SW15 is to hold communication between the microcontroller and RAM module. The RB0 (RAM-CS) switch on the DIP switch SW14 is used to control the access to RAM.

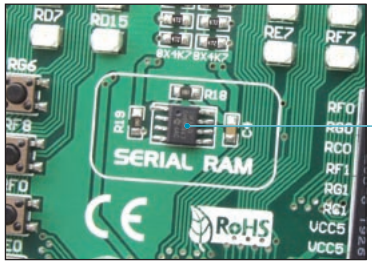


Figure 14-1: Serial RAM module

Serial RAM is connected to the microcontroller via RB14, RB13, RB12, RB0 ans RB1 pins

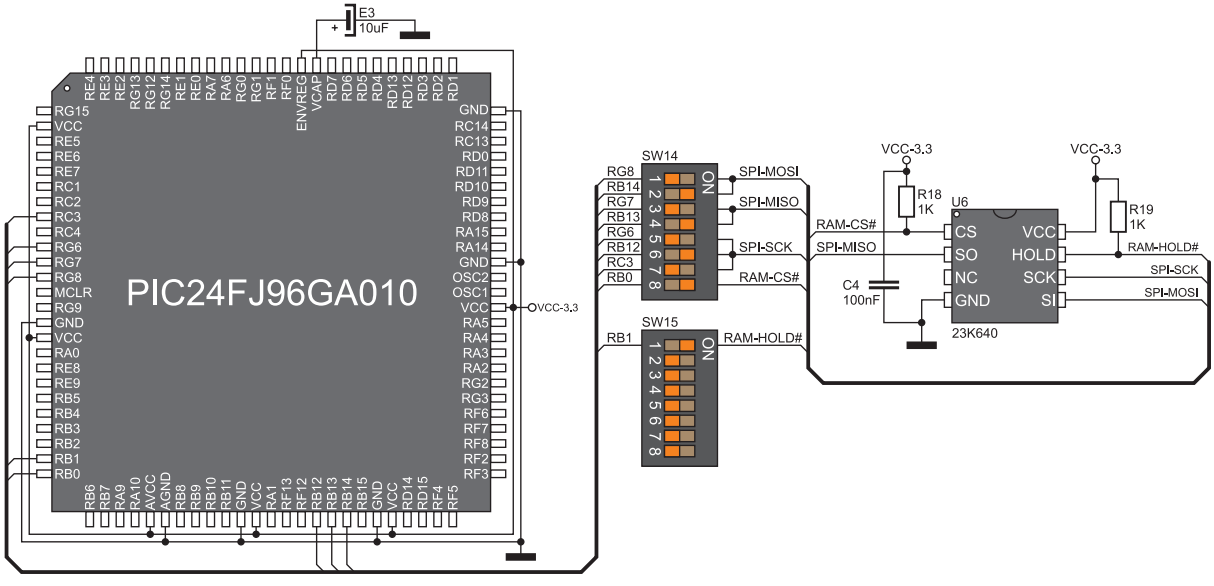


Figure 14-2: Serial RAM connection schematic

page

15.0. LEDs

LED (Light-Emitting Diode) is a highly efficient electronic light source. When connecting LEDs, it is necessary to use a current limiting resistor. A common LED voltage is approximately 2.5V, while the current varies from 1 to 20mA depending on the type of LED. The *LV24-33 v6* uses LEDs with current I=1mA.

There are 96 LEDs on the *LV24-33 v6* development system which visually indicate the state of each microcontroller I/O pin. An active LED indicates that a logic one (1) is present on the pin. In order to enable the pin state to be shown, it is necessary to select appropriate port (PORTA, PORTB, PORTC, PORTD, PORTE, PORTF, PORTF/G or PORTG) using the DIP switch SW13.

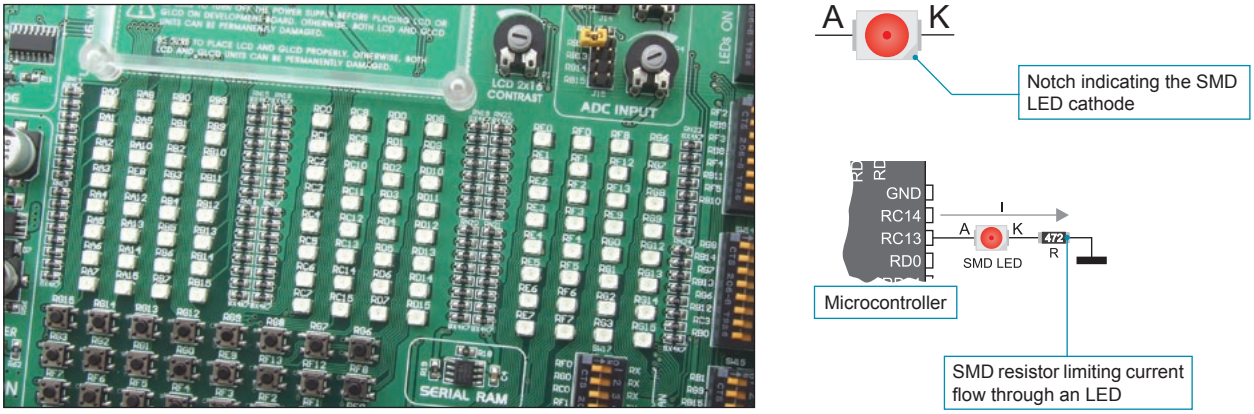


Figure 15-1: LEDs

Port PORTB LEDs are turned on

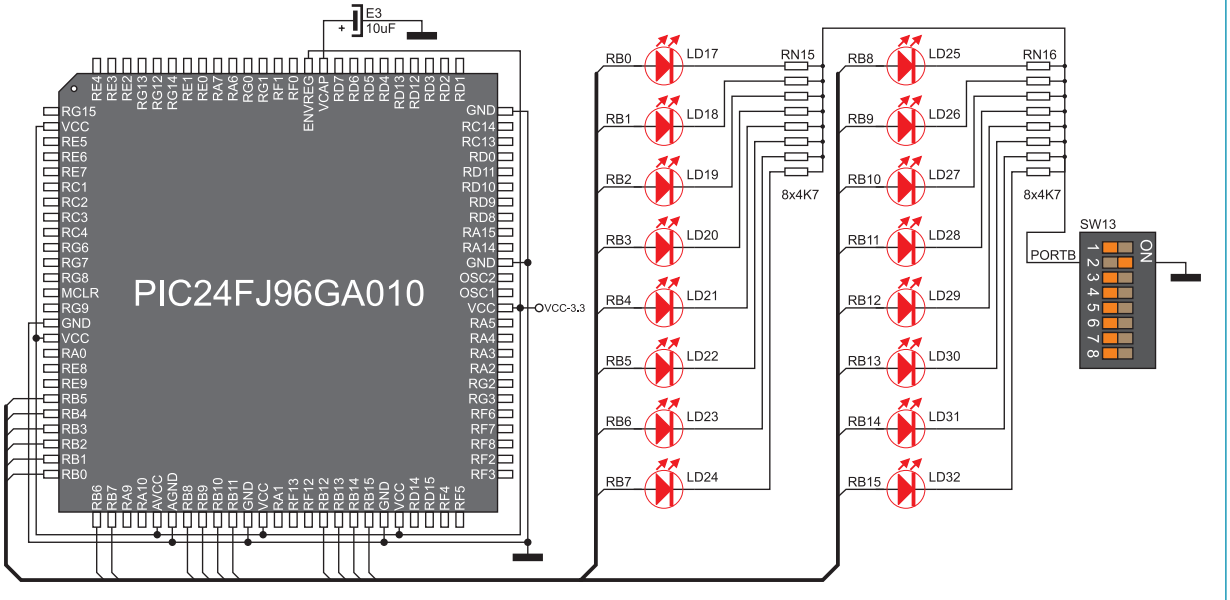


Figure 15-2: LEDs and port PORTB connection schematic

16.0. Push Buttons

The logic state of all microcontroller input pins may be changed by means of push buttons. Jumper J17 is used to determine the logic state to be applied to the desired microcontroller pin by pressing appropriate push button. The function of the protective resistor is to limit the maximum current, thus preventing the development system and peripheral modules from being damaged in case a short circuit occurs. If needed, advanced users may shorten such resistor using jumper J13. Right next to the push buttons, there is a RESET button which is used to provide the MCLR pin with the microcontroller reset signal over the on-board programmer.

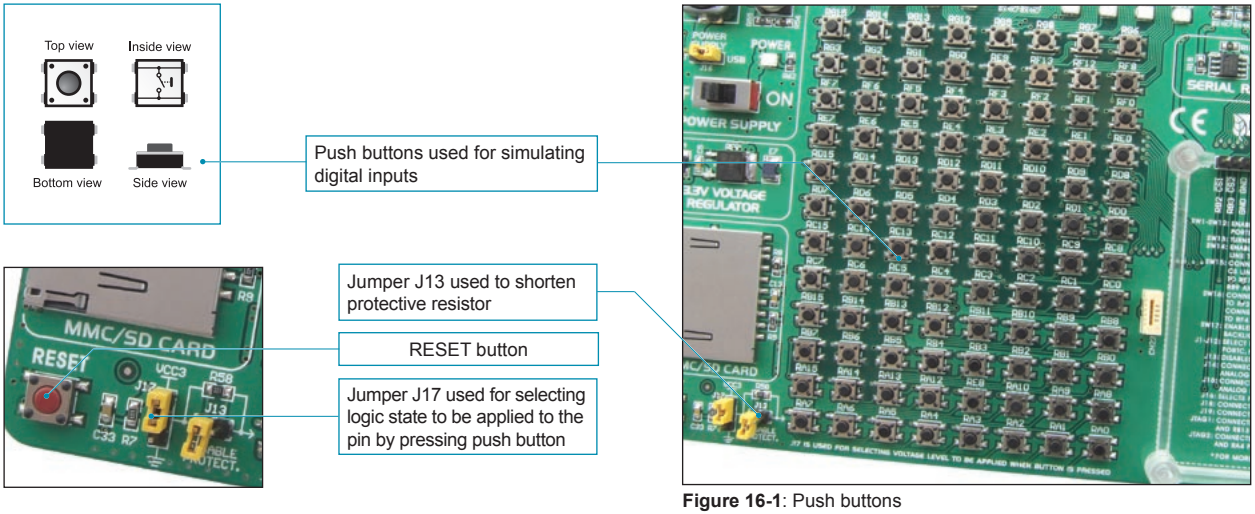

Top view
Inside view
Bottom view
Side view
Push buttons used for simulating digital inputs
Jumper J13 used to shorten protective resistor
RESET button
Jumper J17 used for selecting logic state to be applied to the pin by pressing push button
MMC/SD CARD
RESET
POWER SUPPLY
3.3V VOLTAGE REGULATOR

Figure 16-1: Push buttons

By pressing any push button when jumper J17 is in the VCC-3.3 position, a logic one (3.3V) will be applied to the appropriate microcontroller pin, as shown in Figure 16-2.

By pressing a push button, the appropriate pin will be driven high (1)

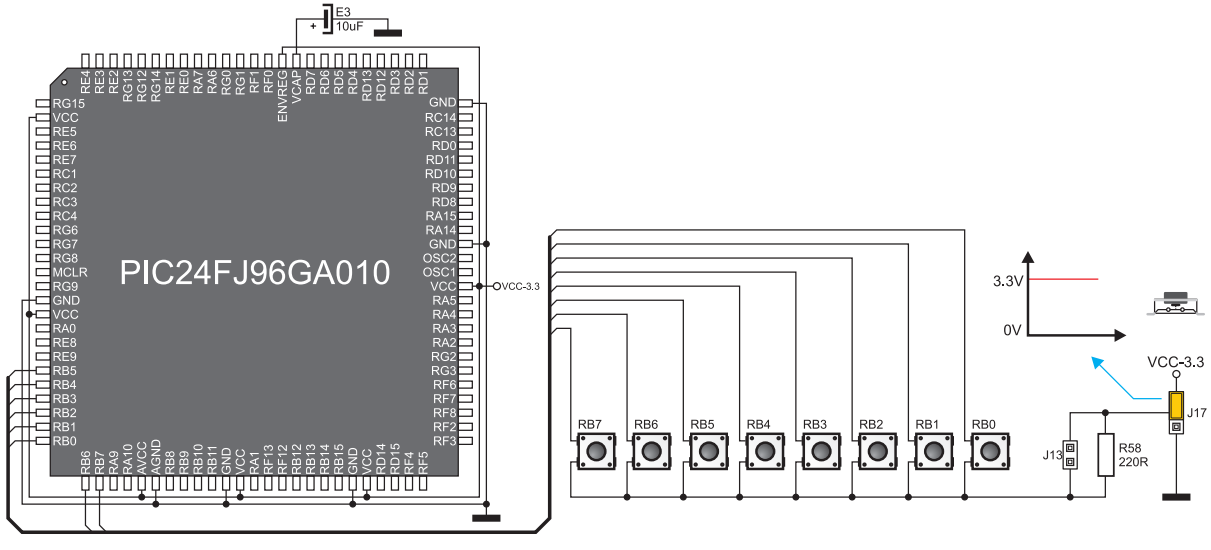


Figure 16-2: Push buttons and port PORTB connection schematic

page

17.0. 2x16 LCD

The *LV24-33 v6* development system provides an on-board connector for the alphanumeric 2x16 LCD. This connector is linked to the microcontroller via port PORTB. Potentiometer P1 is used to adjust display contrast. Switch 7 (LCD-BCK) on the DIP switch SW17 is used to turn the display backlight on/off.
Communication between the LCD and the microcontroller is performed in a 4-bit mode. Alphanumeric digits are displayed in two lines each containing up to 16 characters of 7x5 pixels.

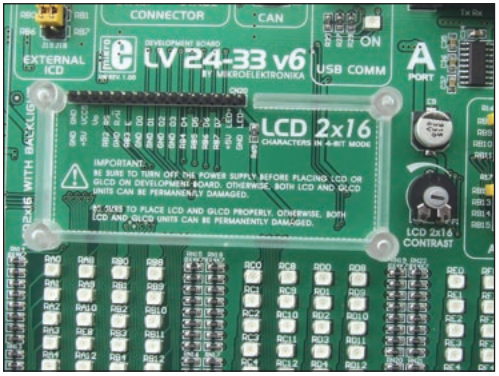


Figure 17-1: Alphanumeric LCD connector

Figure 17-2: Alphanumeric 2x16 LCD

LCD display backlight is on

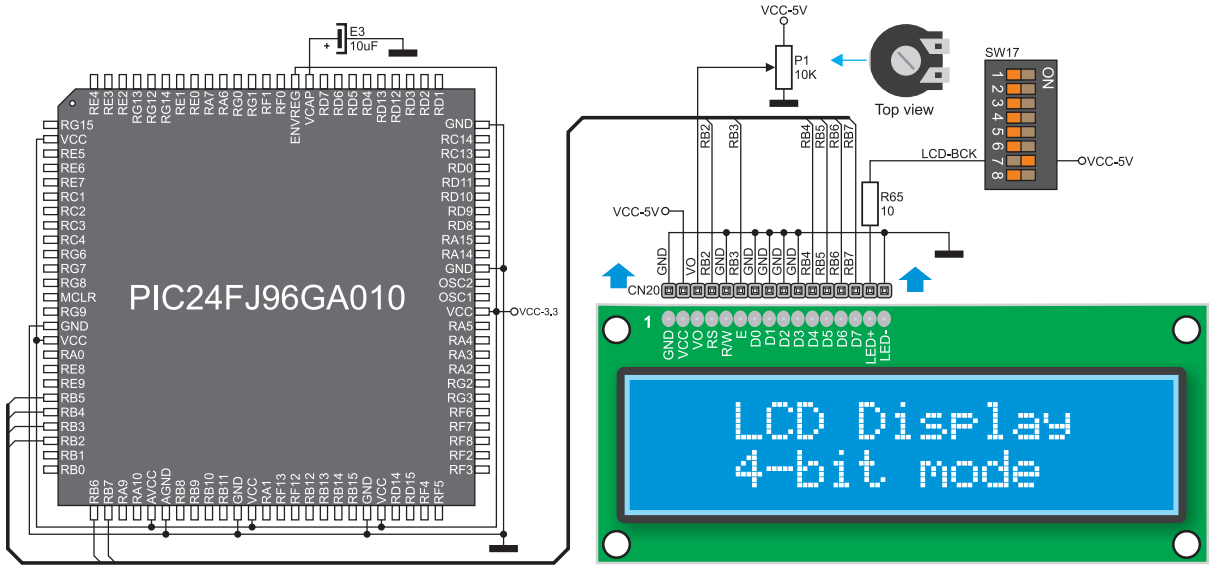


Figure 17-3: Alphanumeric 2x16 LCD connection schematic

18.0. 128x64 Graphic LCD

128x64 graphic LCD (GLCD) is connected to the microcontroller via PORTB and PORTD ports and enables graphic content to be displayed. It has the screen resolution of 128x64 pixels, which allows diagrams, tables and other graphic content to be displayed. Potentiometer P2 is used for the GLCD display contrast adjustment. Switch 8 (GLCD-BCK) on the DIP switch SW17 is used to turn the display backlight on/off. Port PORTB is also used for the operation of 2x16 LCD so that the displays cannot be used simultaneously.

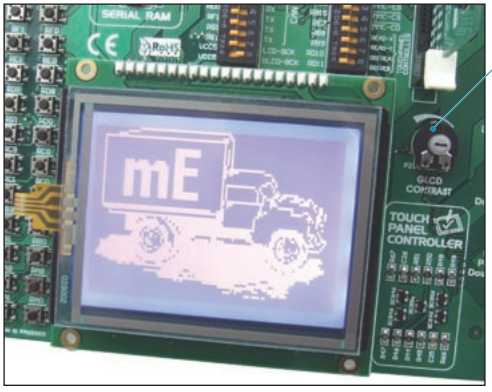


Figure 18-1: GLCD

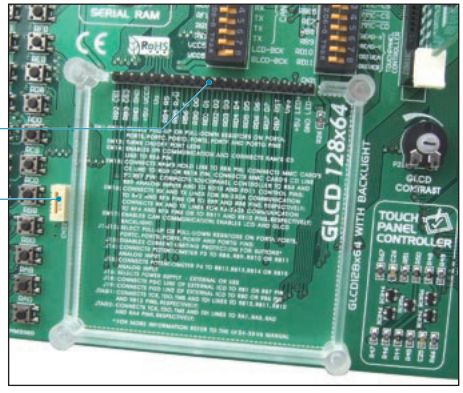


Figure 18-2: GLCD connector

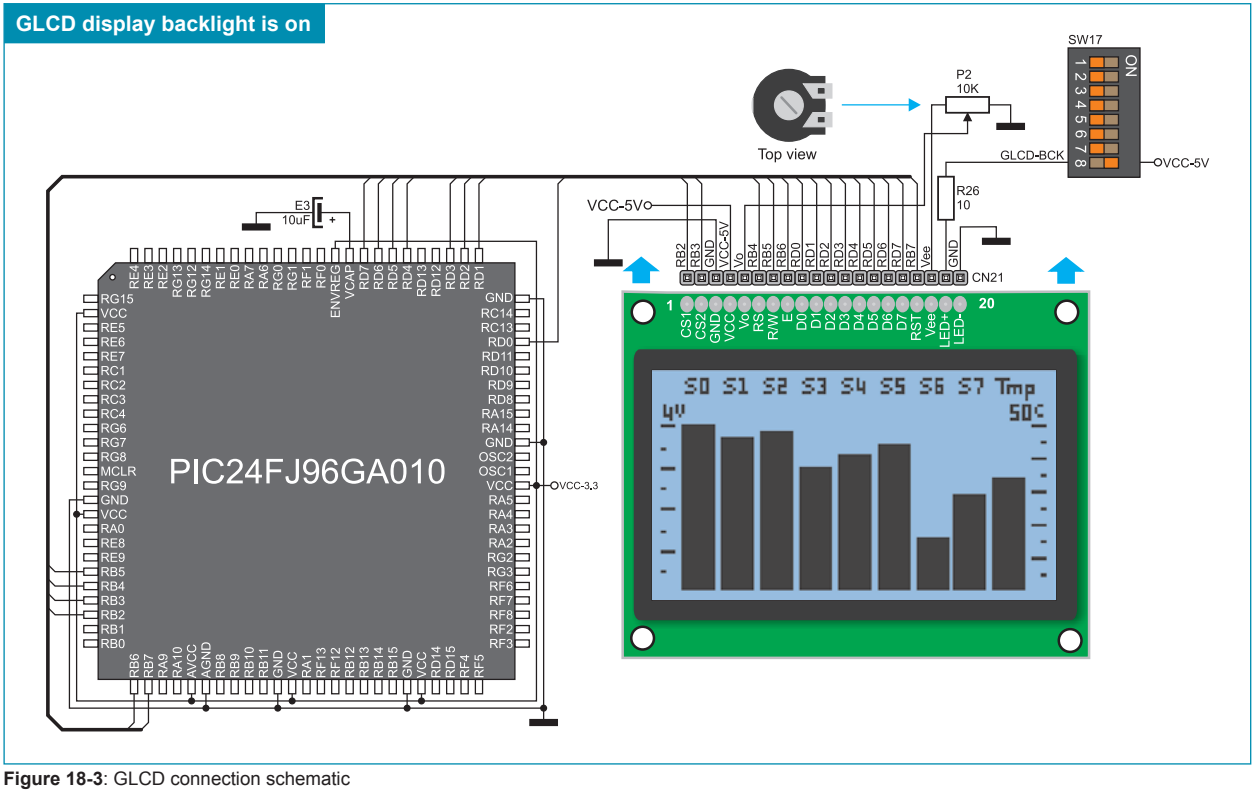


Figure 18-3: GLCD connection schematic

page

19.0. Touch Panel

The touch panel is a thin, self-adhesive, transparent, touch-sensitive panel. It is placed over a GLCD display. Its main function is to register pressure at some specific display point and to forward its coordinates in the form of analog voltage to the microcontroller. Switches 5, 6, 7 and 8 on the DIP switch SW15 are used for connecting the microcontroller and touch panel.

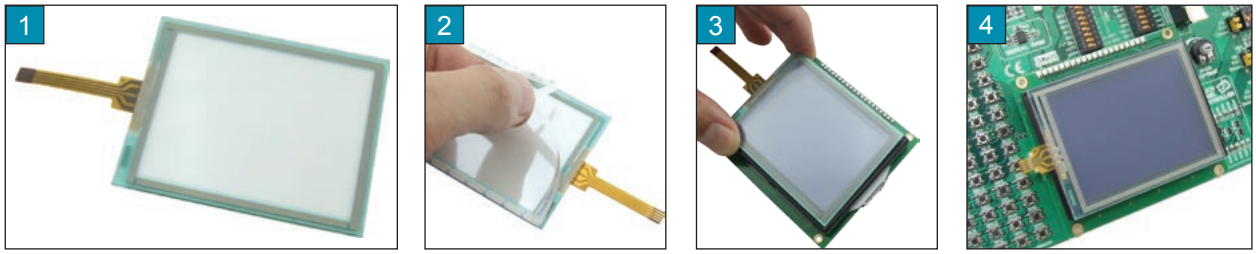


Figure 19-1: Placing touch panel over a GLCD

Figure 19-1 shows how to place a touch panel over a GLCD display. Make sure that the flat cable is to the left of the GLCD, as shown in Figure 4.

Touch panel is connected to the microcontroller via pins RB8, RB9, RD10 and RD11

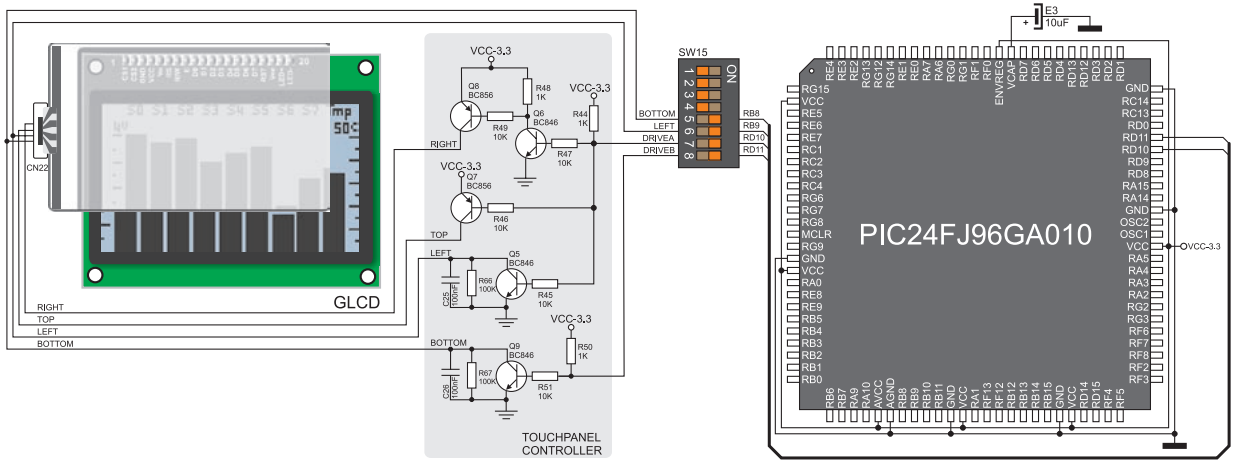


Figure 19-2: Touch panel connection schematic

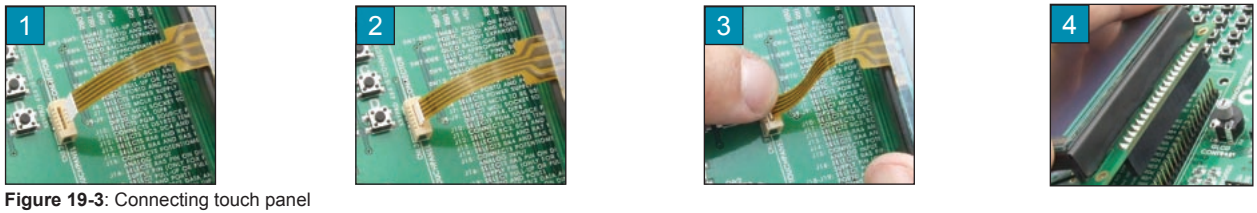


Figure 19-3: Connecting touch panel

Figure 19-3 shows in detail how to connect a touch panel to the microcontroller. Bring the end of the flat cable close to the CN22 connector (Figure 1). Plug the cable into the connector (Figure 2) and press it easily so as to fully fit the connector (Figure 3). Now, a GLCD can be plugged into the appropriate connector (Figure 4).

NOTE: LEDs and pull-up/pull-down resistors on ports PORTB and PORTD must be off when using a touch panel.

20.0. Input/Output Ports

Along the right side of the development system, there are twelve 10-pin connectors connected to the microcontroller's I/O ports. Pins RB0, RB1, RB6, RB7 and MCLR are used for programming and therefore are not directly connected to the appropriate 10-pin connectors, but via the programmer's multiplexer. DIP switch SW1-SW12 enable each connector pin to be connected to one pull-up/pull-down resistor. Whether pins of some port are to be connected to a pull-up or a pull-down resistor depends on the position of jumpers J1-J12.

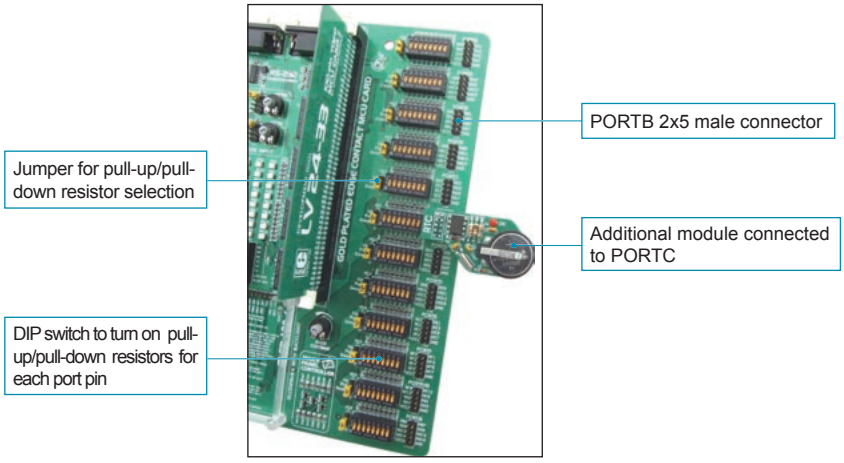

PORTB 2x5 male connector
Jumper for pull-up/pull-down resistor selection
Additional module connected to PORTC
DIP switch to turn on pull-up/pull-down resistors for each port pin

Figure 20-1: I/O ports

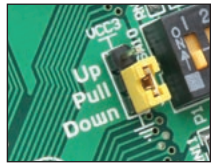


Figure 20-2: J10 in pull-down position

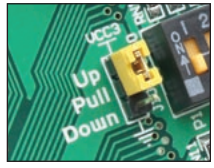


Figure 20-3: J10 in pull-up position

Port PORTB pins are connected to pull-down resistors

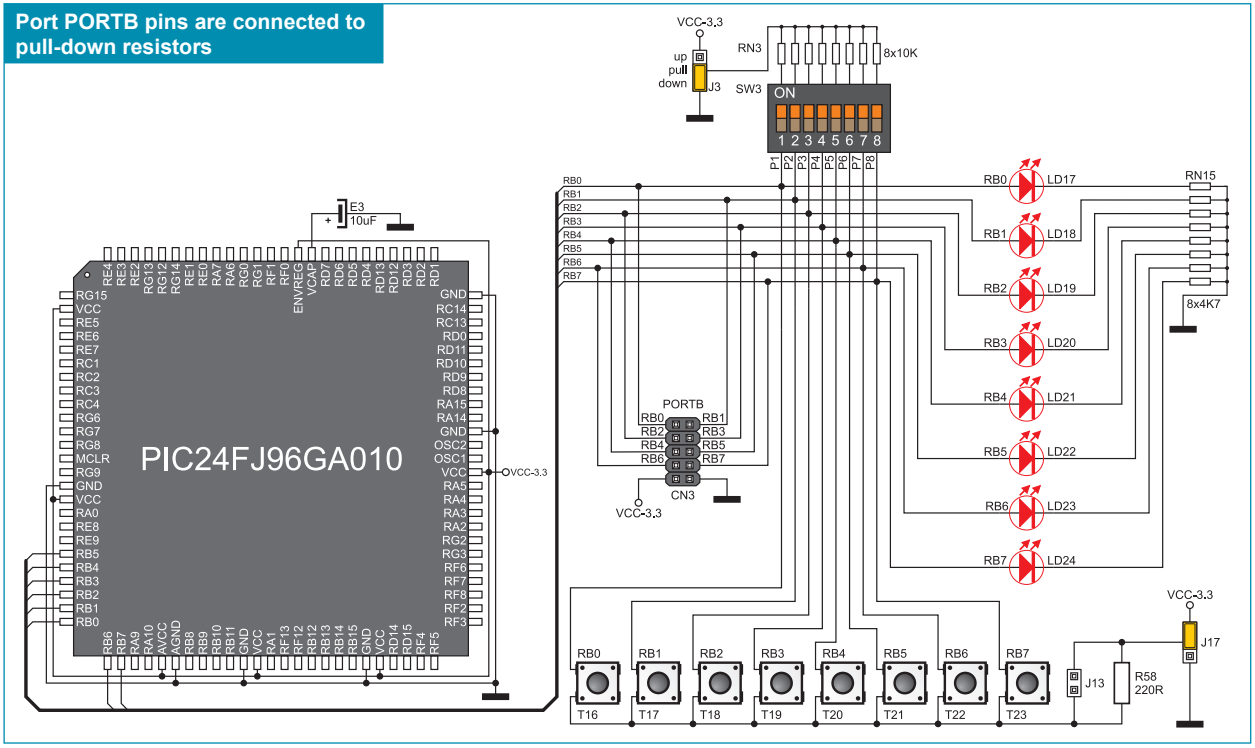


Figure 20-4: Port PORTB connection schematic

page

Pull-up/pull-down resistors enable you to set the logic level on all microcontroller's input pins when they are in idle state. Such level depends on the position of the pull-up/pull-down jumper. The RD8 pin with the relevant DIP switch SW8, jumper J8 and RD8 push button with jumper J17 are used here for the purpose of explaining the performance of pull-up/pull-down resistors. The principle of their operation is the same as for all other microcontroller pins.

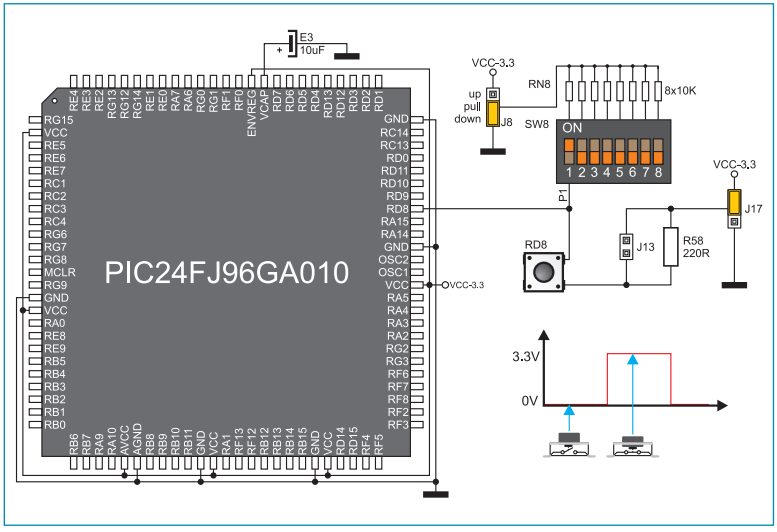


Figure 20-5: Jumper J8 in pull-down and jumper J17 in pull-up position

In order to enable the PORTD port pins to be connected to pull-down resistors, it is necessary to place jumper J8 in the *Down* position first. This enables any PORTD port pin to be supplied with a logic zero (0V) in idle state over jumper J8 and 8x10k resistor network. To provide the RD8 pin with such signal, it is necessary to set switch P1 on the DIP switch SW8 to ON position.

As a result, every time you press the RD8 push button, a logic one (VCC-3.3 voltage) will appear on the PG2 pin, provided that jumper J17 is placed in the VCC-3.3 position.

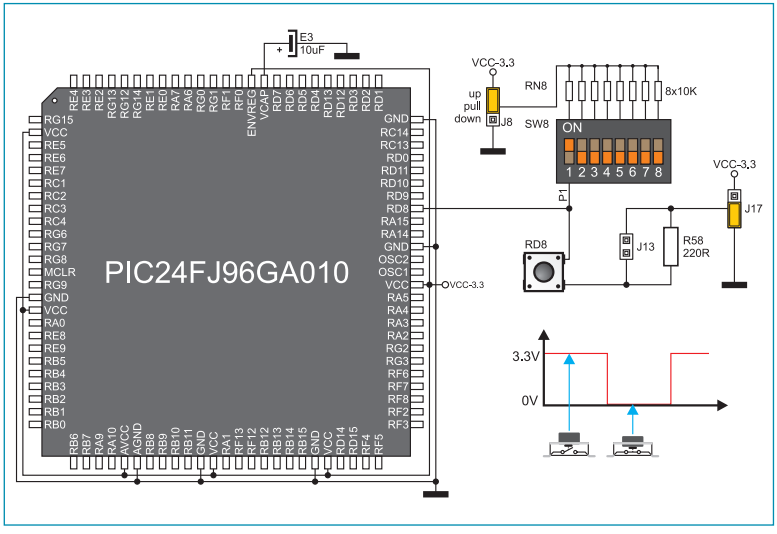


Figure 20-6: Jumper J8 in pull-up and jumper J17 in pull-down position

In order to enable port PORTD pins to be connected to pull-up resistors and the port input pins to be supplied with a logic zero (0), it is necessary to place jumper J8 in the *Up* position and jumper J17 in the GND position. This enables any port PORTD input pin to be driven high (3.3V) in idle state over the 10k resistor.

As a result, every time you press the RD8 push button, a logic zero (0V) will appear on the RD8 pin, provided that the P1 switch is set to the ON position.

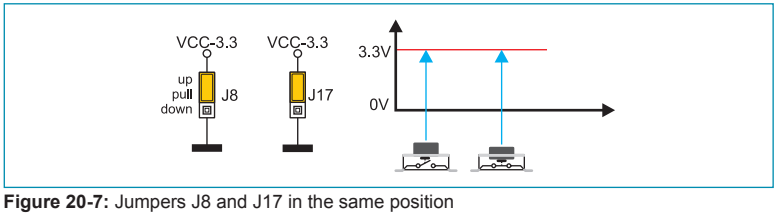


Figure 20-7: Jumpers J8 and J17 in the same position

In case that jumpers J8 and J17 have the same logic state, pressure on any button will not cause input pins to change their logic state.

DISCLAIMER

All the products owned by MikroElektronika are protected by copyright law and international copyright treaty. Therefore, this manual is to be treated as any other copyright material. No part of this manual, including product and software described herein, may be reproduced, stored in a retrieval system, translated or transmitted in any form or by any means, without the prior written permission of MikroElektronika. The manual PDF edition can be printed for private or local use, but not for distribution. Any modification of this manual is prohibited.

MikroElektronika provides this manual 'as is' without warranty of any kind, either expressed or implied, including, but not limited to, the implied warranties or conditions of merchantability or fitness for a particular purpose.

MikroElektronika shall assume no responsibility or liability for any errors, omissions and inaccuracies that may appear in this manual. In no event shall MikroElektronika, its directors, officers, employees or distributors be liable for any indirect, specific, incidental or consequential damages (including damages for loss of business profits and business information, business interruption or any other pecuniary loss) arising out of the use of this manual or product, even if MikroElektronika has been advised of the possibility of such damages. MikroElektronika reserves the right to change information contained in this manual at any time without prior notice, if necessary.

HIGH RISK ACTIVITIES

The products of MikroElektronika are not fault – tolerant nor designed, manufactured or intended for use or resale as on – line control equipment in hazardous environments requiring fail – safe performance, such as in the operation of nuclear facilities, aircraft navigation or communication systems, air traffic control, direct life support machines or weapons systems in which the failure of Software could lead directly to death, personal injury or severe physical or environmental damage ('High Risk Activities'). MikroElektronika and its suppliers specifically disclaim any expressed or implied warranty of fitness for High Risk Activities.

TRADEMARKS

The Mikroelektronika name and logo, the Mikroelektronika logo, mikroC, mikroC PRO, mikroBasic, mikroBasic PRO, mikroPascal, mikroPascal PRO, AVRflash, PICflash, dsPICprog, 18FJprog, PSOCprog, AVRprog, 8051prog, ARMflash, EasyPIC5, EasyPIC6, BigPIC5, BigPIC6, dsPIC PRO4, Easy8051B, EasyARM, EasyAVR5, EasyAVR6, BigAVR2, EasydsPIC4A, EasyPSoC4, EasyVR Stamp LV18FJ, LV24-33A, LV32MX, PIC32MX4 MultiMedia Board, PICPLC16, PICPLC8 PICPLC4, SmartGSM/GPRS, UNI-DS are trademarks of Mikroelektronika. All other trademarks mentioned herein are property of their respective companies.

All other product and corporate names appearing in this manual may or may not be registered trademarks or copyrights of their respective companies, and are only used for identification or explanation and to the owners' benefit, with no intent to infringe.

©Mikroelektronika™, 2010, All Rights Reserved.

If you want to learn more about our products, please visit our website at www.mikroe.com

If you are experiencing some problems with any of our products or just need additional information, please place your ticket at www.mikroe.com/en/support

If you have any questions, comments or business proposals, do not hesitate to contact us at office@mikroe.com

Mouser Electronics

Authorized Distributor

Click to View Pricing, Inventory, Delivery & Lifecycle Information:

mikroElektronika:
MIKROE-468